Files

- AllIn.java
- buff.txt
- charStack.java
- intStack.java
- Main.java
- myQueue.java

CPU RAM Storage

Main.java

```java
import java.util.Scanner;

import java.io.*;

public class Main extends AllIn {
  String fname;

    public Main() {
      System.out.println("Constructor");
      getFileName();
      readFileContents();
    }
        public void getFileName()
    {
      Scanner in = new Scanner(System.in);

      System.out.println("Enter file name please.");
      fname = in.nextLine();
      System.out.println("You entered "+fname);
    }
    public void readFileContents()
    {
      boolean looping;
      DataInputStream in;
      String line;
      int j, len;
      char ch;

    /* Read input from file and process. */
      try {
          in = new DataInputStream(new FileInputStream(fname));

          looping = true;
          while(looping) {
              /* Get a line of input from the file. */
              if (null == (line = in.readLine())) {
                  looping = false;
                  /* Close and free up system resource. */
                  in.close();
              }
              else {
                AllIn val = new AllIn();
                //Validation
                if( line.equals("parentheses tests")){
                  System.out.println("\n" + line + "\n");
                    line = in.readLine();

                  while(!line.equals("postfix equation solving")){

                    System.out.println("\n" + line + "\n"); System.out.println((val.Validation(line)));

                    line = in.readLine();
                    System.out.println("\n");
                  }
                }
                // Evaluation
                System.out.println("**********");
                if( line.equals("postfix equation solving")){
```

Console Shell

```
> sh -c javac -classpath .:target/dependency/ -d . $(find . -type f -name '*.java')
Note: ./Main.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
> java -classpath .:target/dependency/* Main
Hello TV land!
Constructor
Enter file name please.
buff.txt
You entered buff.txt

parentheses tests


((1+2)*3

Invalid: odd number of brackets



(1+2))*3

Invalid: odd number of brackets



y=(z*j)/(b*8)+2

Valid: even number of brackets


**********

postfix equation solving

line = 73+9*
= 90


line = 52+19**
= 63


line = 89*35+-
= 64


line = 21+84/+
= 5


**********

infix to postfix conversion

line = (8+4)*9
= 8 4 + 9 *


line = (6+3)*(2*8)
= 6 3 + 2 8 * *


line = (9*8)-(5+3)
= 9 8 * 5 3 + -


line = 3+(9/3)
```

Files

- AllIn.java
- buff.txt
- charStack.java
- intStack.java
- Main.java
- myQueue.java

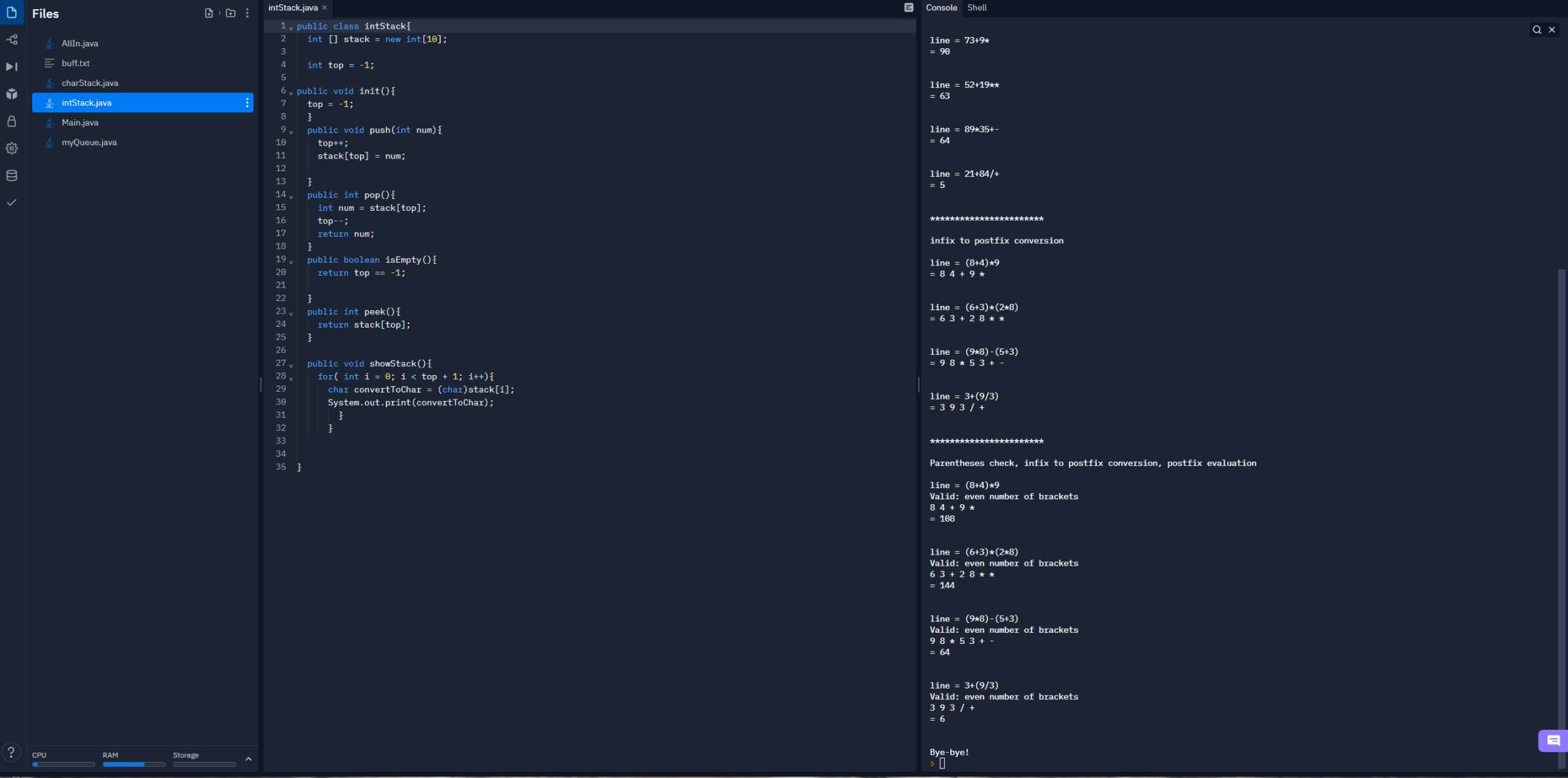
CPU RAM Storage

charStack.java

```java

class charStack {
  int top;
  char[] charStack = new char[100];
  char c;

  public void init()
  {
    top = -1;
  }

  public void push(char item)
  {
    top++;
    charStack[top] = item;
  }

  public char pop()
  {

    char c;
    c = charStack[top];
    top--;
    return c;
  }

  public boolean isEmpty()
  {
    if(top == -1){
      return true;
    }
    else return false;
  }

  public int getTop()
  {
    return charStack[top];
  }

  public void showStack()
  {
    int j;
    for(j=0;j<=top;j++){
      System.out.println(charStack[j]);
        }
    }
}

```

Console Shell

```
line = 73+9*
= 90


line = 52+19**
= 63


line = 89*35+-
= 64


line = 21+84/+
= 5


**********

infix to postfix conversion

line = (8+4)*9
= 8 4 + 9 *


line = (6+3)*(2*8)
= 6 3 + 2 8 * *


line = (9*8)-(5+3)
= 9 8 * 5 3 + -


line = 3+(9/3)
= 3 9 3 / +


**********

Parentheses check, infix to postfix conversion, postfix evaluation

line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108


line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144


line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64


line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6


Bye-bye!
```


Files
AllIn.java
buff.txt
charStack.java
intStack.java
Main.java
myQueue.java
CPU
RAM
Storage
intStack.java
1 public class intStack{
2 int [] stack = new int[10];
3
4 int top = -1;
5
6 public void init(){
7 top = -1;
8 }
9 public void push(int num){
10 top++;
11 stack[top] = num;
12
13 }
14 public int pop(){
15 int num = stack[top];
16 top--;
17 return num;
18 }
19 public boolean isEmpty(){
20 return top == -1;
21
22 }
23 public int peek(){
24 return stack[top];
25 }
26
27 public void showStack(){
28 for(int i = 0; i < top + 1; i++){
29 char convertToChar = (char)stack[i];
30 System.out.print(convertToChar);
31 }
32 }
33
34
35 }
Console
Shell
line = 73+9*
= 90
line = 52+19**
= 63
line = 89*35+-
= 64
line = 21+84/+
= 5
****
infix to postfix conversion
line = (8+4)*9
= 8 4 + 9 *
line = (6+3)*(2*8)
= 6 3 + 2 8 * *
line = (9*8)-(5+3)
= 9 8 * 5 3 + -
line = 3+(9/3)
= 3 9 3 / +
****
Parentheses check, infix to postfix conversion, postfix evaluation
line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108
line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144
line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64
line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6
Bye-bye!

- AllIn.java
- buff.txt
- charStack.java
- intStack.java
- Main.java
- myQueue.java

AllIn.java

```
    symbol = eq.charAt(i);

  if(symbol == '[' || symbol == '(' || symbol == '{'){
    s.push(symbol);

}
    else if(symbol == ']' || symbol == ')' || symbol == '}'){
      if(s.isEmpty() == false){
        s.pop();
      }
      else answer = false;
  }
    } //Ends for loop
  if(!s.isEmpty()){
  answer = false;
  }
  if(answer == true){
    return "Valid: even number of brackets";
  }
  else return "Invalid: odd number of brackets";
    }

  //Translation
  public static String Translate (String eq) {

      String translatedString = "";
      myQueue q = new myQueue();
      intStack s = new intStack();
      q.init();
      s.init();
      char op;
      for (int i = 0; i < eq.length(); ++i) {
          char[] ch = eq.toCharArray();
          if (Character.isLetterOrDigit(ch[i])) {
              q.pushQ(ch[i]);
          }
          else if (ch[i] == '-' ||  ch[i] == '*' || ch[i] == '+' || ch[i] == '/' ) {
              s.push(ch[i]);
              }

          else if (ch[i] == ')') {
              op = (char)(s.pop());
                  q.pushQ(op);
              }

      }//Ends for loop
      while (!(s.isEmpty())) {
          op = (char)(s.pop());
          q.pushQ(op);
      }

  for(int i = 0; i< q. myQueue.length; i++) {

      translatedString = translatedString + q.myQueue[i]+ " ";
      }

  return translatedString;
    }
```

Console Shell

```
line = 73+9*
= 90


line = 52+19**
= 63


line = 89*35+-
= 64


line = 21+84/+
= 5


**********

infix to postfix conversion

line = (8+4)*9
= 8 4 + 9 *


line = (6+3)*(2*8)
= 6 3 + 2 8 * *


line = (9*8)-(5+3)
= 9 8 * 5 3 + -


line = 3+(9/3)
= 3 9 3 / +


**********

Parentheses check, infix to postfix conversion, postfix evaluation

line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108


line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144


line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64


line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6


Bye-bye!
```

CPU RAM Storage

Files

- AllIn.java
- buff.txt
- charStack.java
- intStack.java
- Main.java
- myQueue.java

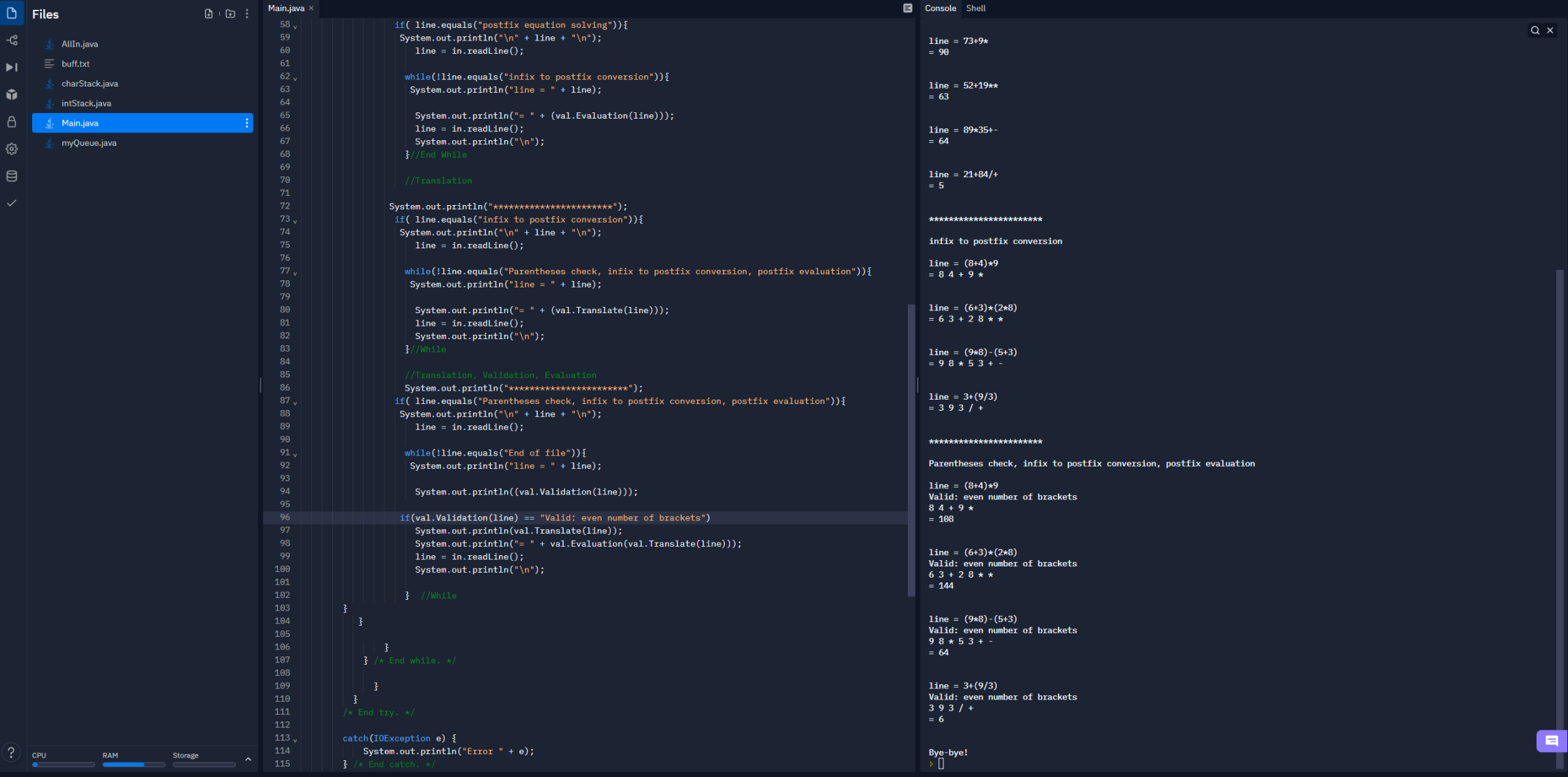
myQueue.java

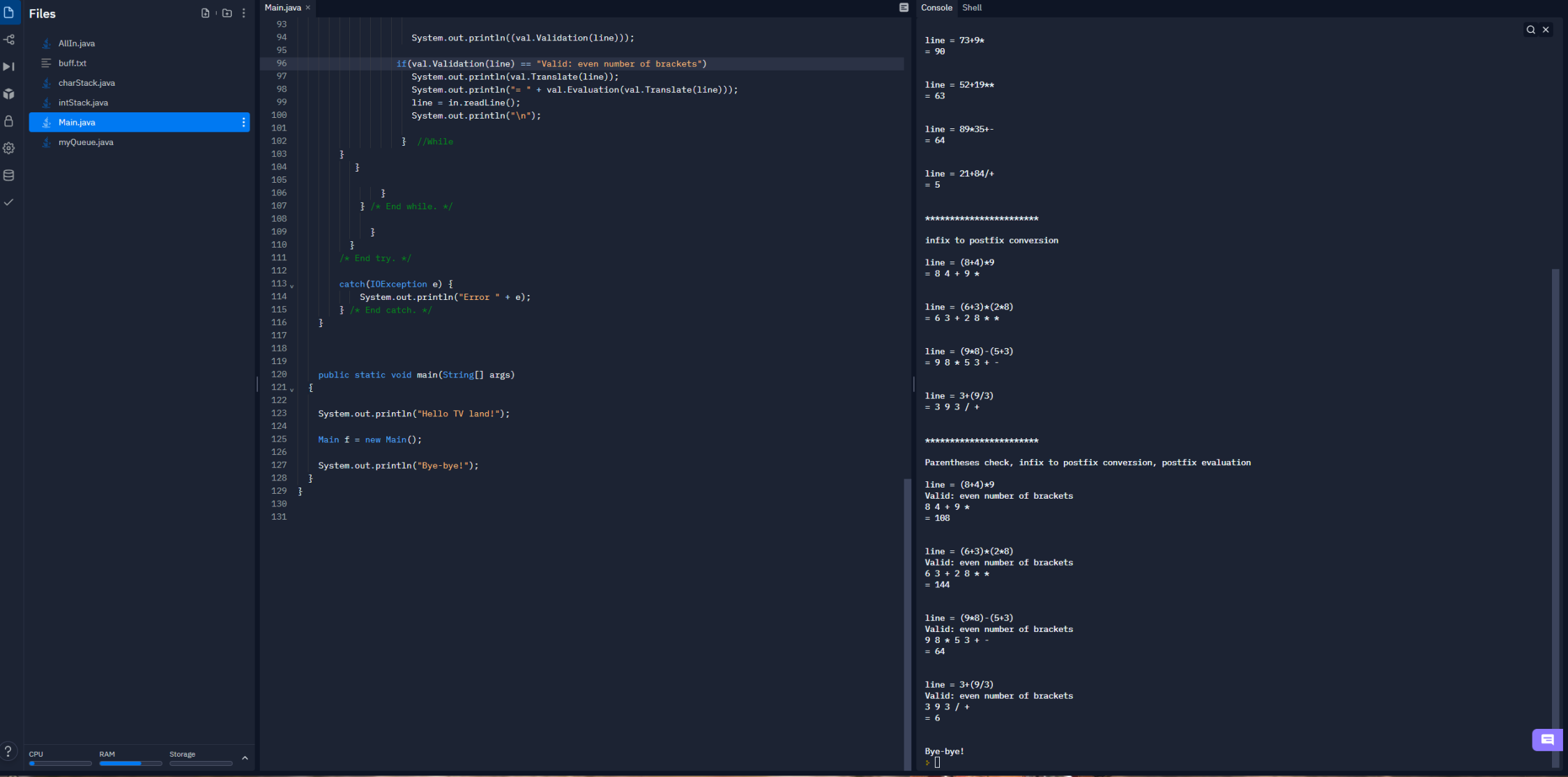
```java
public class myQueue {

  int front;
  int rear;

  char[] myQueue = new char[10];

  public void init() {
    front = 0;
    rear = -1;

  }

  public void pushQ(char x) {
      rear = rear+1;
      myQueue[rear] = x;
  }

  int popQ() {
      char x;
      x = myQueue[front];
      front = front+1;
      return x;
  }


  public boolean isEmpty() {
      boolean empty;
      empty = false;
      if(front>=rear) {
          empty = true;
      }
      return empty;
      }


  }
```

Console Shell

```
line = 73+9*
= 90


line = 52+19**
= 63


line = 89*35+-
= 64


line = 21+84/+
= 5


**********

infix to postfix conversion

line = (8+4)*9
= 8 4 + 9 *


line = (6+3)*(2*8)
= 6 3 + 2 8 * *


line = (9*8)-(5+3)
= 9 8 * 5 3 + -


line = 3+(9/3)
= 3 9 3 / +


**********

Parentheses check, infix to postfix conversion, postfix evaluation

line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108


line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144


line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64


line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6


Bye-bye!
```

CPU RAM Storage


Files
AllIn.java
buff.txt
charStack.java
intStack.java
Main.java
myQueue.java
CPU
RAM
Storage
Main.java
if(line.equals("postfix equation solving")){
System.out.println("\n" + line + "\n");
line = in.readLine();
while(!line.equals("infix to postfix conversion")){
System.out.println("line = " + line);
System.out.println("= " + (val.Evaluation(line)));
line = in.readLine();
System.out.println("\n");
}//End While
//Translation
System.out.println("**********");
if(line.equals("infix to postfix conversion")){
System.out.println("\n" + line + "\n");
line = in.readLine();
while(!line.equals("Parentheses check, infix to postfix conversion, postfix evaluation")){
System.out.println("line = " + line);
System.out.println("= " + (val.Translate(line)));
line = in.readLine();
System.out.println("\n");
}//While
//Translation, Validation, Evaluation
System.out.println("**********");
if(line.equals("Parentheses check, infix to postfix conversion, postfix evaluation")){
System.out.println("\n" + line + "\n");
line = in.readLine();
while(!line.equals("End of file")){
System.out.println("line = " + line);
System.out.println((val.Validation(line)));
if(val.Validation(line) == "Valid: even number of brackets")
System.out.println(val.Translate(line));
System.out.println("= " + val.Evaluation(val.Translate(line)));
line = in.readLine();
System.out.println("\n");
} //While
}
}
} /* End while. */
}
/* End try. */
catch(IOException e) {
System.out.println("Error " + e);
} /* End catch. */
Console
Shell
line = 73+9*
= 90
line = 52+19**
= 63
line = 89*35+-
= 64
line = 21+84/+
= 5
****
infix to postfix conversion
line = (8+4)*9
= 8 4 + 9 *
line = (6+3)*(2*8)
= 6 3 + 2 8 * *
line = (9*8)-(5+3)
= 9 8 * 5 3 + -
line = 3+(9/3)
= 3 9 3 / +
****
Parentheses check, infix to postfix conversion, postfix evaluation
line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108
line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144
line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64
line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6
Bye-bye!


Files
AllIn.java
buff.txt
charStack.java
intStack.java
Main.java
myQueue.java
CPU
RAM
Storage
Main.java
System.out.println((val.Validation(line)));
if(val.Validation(line) == "Valid: even number of brackets")
System.out.println(val.Translate(line));
System.out.println("= " + val.Evaluation(val.Translate(line)));
line = in.readLine();
System.out.println("\n");
} //While
} /* End while. */
/* End try. */
catch(IOException e) {
System.out.println("Error " + e);
} /* End catch. */
public static void main(String[] args)
{
System.out.println("Hello TV land!");
Main f = new Main();
System.out.println("Bye-bye!");
}
}
Console
Shell
line = 73+9*
= 90
line = 52+19**
= 63
line = 89*35+-
= 64
line = 21+84/+
= 5
****
infix to postfix conversion
line = (8+4)*9
= 8 4 + 9 *
line = (6+3)*(2*8)
= 6 3 + 2 8 * *
line = (9*8)-(5+3)
= 9 8 * 5 3 + -
line = 3+(9/3)
= 3 9 3 / +
****
Parentheses check, infix to postfix conversion, postfix evaluation
line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108
line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144
line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64
line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6
Bye-bye!

Files

- AllIn.java
- buff.txt
- charStack.java
- intStack.java
- Main.java
- myQueue.java

AllIn.java

```java
public class AllIn{

  //Evaluation
int Evaluation(String eq){
intStack stacked = new intStack();
int num, x, y ,z, j;
char myChar;
stacked.init();
for(j=0; j<eq.length(); j++){
  myChar = eq.charAt(j);
  if((myChar >= '0') && (myChar <= '9')){
    num = myChar - '0';
    stacked.push(num);
  }

  else if(myChar == '+'){
    y = stacked.pop();
    x = stacked.pop();
    z = x + y;
    stacked.push(z);
  }
    else if(myChar == '-'){
    y = stacked.pop();
    x = stacked.pop();
    z = x - y;
    stacked.push(z);
  }
    else if(myChar == '/'){
    y = stacked.pop();
    x = stacked.pop();
    z = x / y;
    stacked.push(z);
  }
    else if(myChar == '*'){
    y = stacked.pop();
    x = stacked.pop();
    z = x * y;
    stacked.push(z);
  }
}//Ends for loop

  return stacked.pop();

}

//Validation

String Validation(String eq){
  int i;
charStack s = new charStack();
int top;
char symbol;
s.init();

  boolean answer;
  answer = true;
  for(i=0; i<eq.length();i++){
    symbol = eq.charAt(i);
```

Console Shell

```
line = 73+9*
= 90


line = 52+19**
= 63


line = 89*35+-
= 64


line = 21+84/+
= 5


**********

infix to postfix conversion

line = (8+4)*9
= 8 4 + 9 *


line = (6+3)*(2*8)
= 6 3 + 2 8 * *


line = (9*8)-(5+3)
= 9 8 * 5 3 + -


line = 3+(9/3)
= 3 9 3 / +


**********

Parentheses check, infix to postfix conversion, postfix evaluation

line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108


line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144


line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64


line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6


Bye-bye!
```

CPU RAM Storage

```
sh -c javac -classpath .:target/dependency/ -d . $(find . -type f -name '*.java')
Note: ./Main.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
java -classpath .:target/dependency/* Main
Hello TV land!
Constructor
Enter file name please.
buff.txt
You entered buff.txt

parentheses tests


((1+2)*3

Invalid: odd number of brackets



(1+2))*3

Invalid: odd number of brackets



y=(z*j)/(b*8)+2

Valid: even number of brackets


**********

postfix equation solving

line = 73+9*
= 90


line = 52+19**
= 63


line = 89*35+-
= 64


line = 21+84/+
= 5


**********

infix to postfix conversion

line = (8+4)*9
= 8 4 + 9 *


line = (6+3)*(2*8)
= 6 3 + 2 8 * *


line = (9*8)-(5+3)
= 9 8 * 5 3 + -
```

```
line = 52+19**
= 63


line = 89*35+-
= 64


line = 21+84/+
= 5


**********

infix to postfix conversion

line = (8+4)*9
= 8 4 + 9 *


line = (6+3)*(2*8)
= 6 3 + 2 8 * *


line = (9*8)-(5+3)
= 9 8 * 5 3 + -


line = 3+(9/3)
= 3 9 3 / +


**********

Parentheses check, infix to postfix conversion, postfix evaluation

line = (8+4)*9
Valid: even number of brackets
8 4 + 9 *
= 108


line = (6+3)*(2*8)
Valid: even number of brackets
6 3 + 2 8 * *
= 144


line = (9*8)-(5+3)
Valid: even number of brackets
9 8 * 5 3 + -
= 64


line = 3+(9/3)
Valid: even number of brackets
3 9 3 / +
= 6


Bye-bye!
```